Sensing Through the Body – Non-Contact Object Localisation Using Morphological Computation

Euan Judd[1,2], Gabor Soter[1,2], Jonathan Rossiter[1,2], Helmut Hauser[1,2]

***Abstract*— Biological systems exhibit remarkable sensing capabilities by using a widely distributed sensor network and a tight coupling between body and environment. The result are seemingly highly robust and adaptive solutions. To enable the next generation of embodied robots with similar capabilities, we need to develop novel sensing and computational technologies. In this paper, we propose an approach using proprioceptive sensing and leveraging the system-environmental interaction in soft robotics based on the principle of morphological computation, i.e. the use of morphological features for computational tasks. We exploit the body dynamics of a moving octopus-inspired robot tentacle in coordination with the dynamics of the surrounding water to predict the position of objects in its vicinity without touching them. The values of proprioceptive strain sensors, which were emulated with the help of computer vision techniques on recorded videos of the experiments, and simple linear regression on these values were sufficient to solve this computational prediction problem. We were able to demonstrate that the body of the soft tentacle could be used to "feel" the location of an object by observing its own body dynamics (strain sensors) which are responding to changes (i.e. different positions of the object) in the environment.**

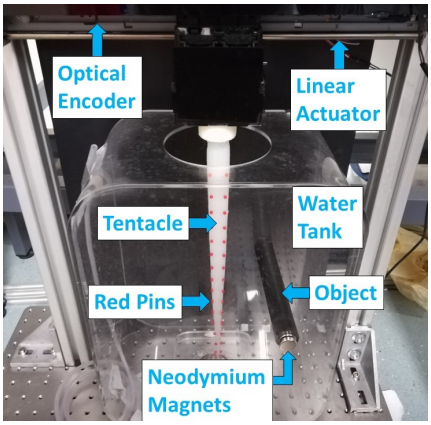


Fig. 1. The octopus inspired EcoFlex® 00-10 silicone tentacle was suspended vertically in water from a linear actuator. 32 red pins, 16 down each side of the tentacle, were used to obtain 30 strain measurements using computer vision techniques. A cylindrical object was suspended in the water far enough from the tentacle so that they never contact.

I. INTRODUCTION

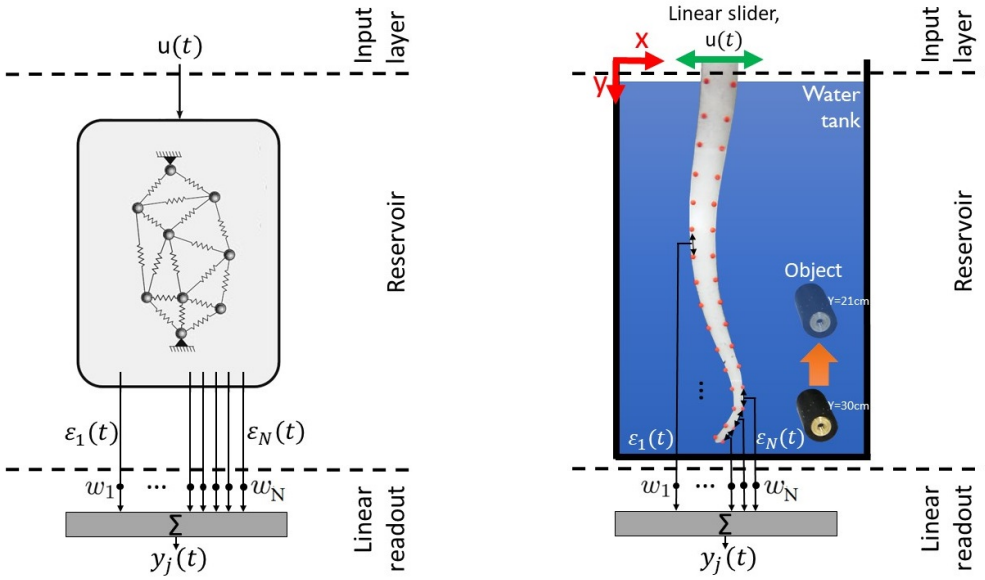
The theory of embodied intelligence suggests that behaviour is an emergent property of the interaction between a system's controller, morphology and environment [1]. This has been exploited in a number of robotic applications. For example, Corucci et al. [2] used this principle to simplify control of a simulated underwater quadrupedal robot allowing it to transition between qualitatively different modes of locomotion by changing a single morphological parameter. Another example is [3], which uses the idea of embodied intelligence for the control of a terrestrial quadrupedal robot. The magnitude of the change in ground friction and the robot's weight were sensed by measuring locomotion stability–i.e. periodicity–of the motor torque, joint angle, acceleration, and foot pressure sensors. It has also been shown that the interaction of morphology and environment can play a crucial role in sensing. For example, it has been demonstrated that the interaction of body and environment is fundamental for the development of the sensorimotor loops [4]. For a general discussion and more examples, we refer to [5].

While these examples are encouraging, little work has been done on formalizing the idea mathematically. One approach has been proposed by Hauser et al. [6], [7] by using nonlinear mass-spring-damper systems to provide general models of how soft bodies can carry out computational tasks for control and sensing. Their theoretical models are based on the machine learning technique called *reservoir computing*, which exploits high-dimensional nonlinear, dynamical systems as a computational resource [8].

In this context, the morphology of the body can be seen as a kernel in the Machine Learning sense. This means the body can provide the nonlinear projection of input variables (e.g. forces from the environment) into its high-dimensional state space (state of the soft body). For a more detailed discussion on kernels, we refer to [6], [8]. The soft body can therefore be represented as a nonlinearly connected mass-spring-damper system which maps low-dimensional input to a high-dimensional output (Fig. 2). Being a dynamical system, it inherently has the property to integrate input information over time which is beneficial for computation which needs information on the history of input values [9]. If we use the body as a reservoir then the addition of linear outputs is all that is required to carry out complex, nonlinear computations. Interestingly, we only need to adapt this linear readout to learn a given computational task. This means, with the help of the body dynamics as a computational resource, we can use simple linear regression to learn complex computational tasks, including nonlinear controllers and nonlinear filtering for sensing.

It has already been shown that a soft silicone tentacle can be used to perform numerous computations using this approach [9], [10]. In this paper we propose to use the body

[1]Department of Engineering Mathematics, University of Bristol, United Kingdom. euan.judd@brl.ac.uk
[2]Bristol Robotics Laboratory, Bristol, United Kingdom.



(a) Physical reservoir

(b) Tentacle as a physical reservoir

Fig. 2. (a) A physical reservoir in reservoir computing consists of mass-spring-damper systems that nonlinearly map a low-dimensional input, $u(t)$, to a high dimensional output, $\boldsymbol{\varepsilon}(t)$. The output of a reservoir, $y_j(t)$, can be calculated as the addition of the linear readouts, $\varepsilon_i(t)$, that have been multiplied by output weights, w_i, where i is the number of a particular readout. Learning is therefore reduced to finding the weights, $\boldsymbol{w}$, the only adapted part of the system, using linear regression. (Diagram adapted from [6].) (b) The silicone tentacle is a physical reservoir where the control input to the linear actuator is the low-dimensional input and strain measurements between red pins are the high-dimensional linear readouts.

of a soft silicone tentacle as a sensor to predict the position of an object in its vicinity without touching it (Fig. 1). The tentacle is moved by a linear actuator and thereby introduces dynamics in the water. The environment (i.e. the water) will respond differently for different positions of the object. In response, these small differences in hydrodynamics in the environment of the tentacle, reciprocally, will also change the state of the soft (passive) tentacle, which can be picked up through proprioceptive sensing. At the same time the body will be used as a nonlinear filter (reservoir), which allows us to employ a simple linear readout to predict the position of the object.

II. METHOD

A. SETUP

An octopus-inspired tentacle was fabricated using EcoFlex® 00-10 silicone (smooth-on.com). The tentacle was 34cm long with a maximum diameter of 3cm at the proximal end, which was attached to a linear slider. The tentacle diameter tapers down to 0.6cm at the distal tip.

Strain sensors were emulated using computer vision techniques in order to measure the behaviour of the tentacle and provide the linear outputs from the reservoir. This contact free method was chosen as it would allow us to have a high-dimensional output from the body while not significantly affecting the computational performance of the tentacle or the complex behaviour induced by environmental changes. Therefore, 32 red pins were placed uniformly along the tentacle, 16 on either side, in such a way that they would be visible to a video camera at all times (Fig. 1). A total of 32 markers provided 30 emulated strain measurements; 15 between successive markers on the left of the tentacle and 15 between successive markers on the right side of the tentacle. This allowed compressive and tensile strains to be recorded on both sides of the tentacle.

The tentacle was suspended vertically from a motorised linear slider (Fig. 1). The stepper motor was driven using an Adafruit Motorshield on an Arduino Uno. The tentacle dangled into an acrylic tank filled with water so that all markers on the tentacle were submerged. The cylindrical object, which should be detected, was of diameter 2.7cm and was moved and secured in the tank using neodymium magnets.

B. DATA ACQUISITION

A Panasonic Lumix 4k DMC-G80 camera was used to record videos at 28 Megapixels and 60 frames per second (FPS) with the 12-60mm lens set to 12mm. Refraction of light through the tank was ignored as reservoir computing is robust enough and does not require highly accurate readouts from a reservoir. The stepper motor was provided with a voltage of 10V and a current of 1A and the desired position, p, of the linear slider was controlled using the following sinusoidal function

$$p = Asin(\omega t), \tag{1}$$

where the amplitude, A, was the distance that the linear slider moves and was measured using an optical encoder, ω was the angular frequency equal to $2\pi f$, where f was the frequency and t was the elapsed time in seconds. The amplitude was provided to the desired position equation as the number of incremental scale periods (each $180\mu m$).

Various experiments were conducted with different amplitudes, frequencies and object positions to systematically investigate the proposed setup and its limitations. Once the videos had been captured, OpenCV [11] was utilised to acquire strain measurements between the 32 red markers. The developed C++ algorithm, which took successive frames as inputs, is described below:

1) The red pins were separated out from the rest of the image using an adaptive threshold that set any pixel with a high red channel and a low blue and green channel to black. Every other pixel was set to white.
2) The resulting RGB image was converted to grayscale. The image was further processed to improve the performance of the blob detector used in the next step. OpenCV's `cv::erode()` removed protrusions while `cv::dilate()` filled concavities. Finally, `cv::blur()` was used to remove noise.
3) OpenCV's `cv::SimpleBlobDetector` was used to determine the centroid coordinates of each pin. The function extracts connected components that have a user specified degree of circularity, area, distance apart from other connected areas, and convexity.
4) Finally, pins were tracked between frames starting from the first frame (Fig. 2b). In this way we could label each pin in the first frame by measuring the distance between the pin and the coordinate origin. When the second frame was processed, in which the stepper motor had moved, the pins were labelled based on how far they were to the nearest pin in the previous frame as measured by Euclidean distance.

Once the pin positions in the first and second frame were found, the strain between pins could be found easily using the following formula

$$\varepsilon = (L - L_0)/L_0, \tag{2}$$

where ε is the strain between pins, and L_0 and L are the distances between pins in the first and second frames respectively.

C. PREDICTIONS

By using the body as a computational resource, i.e. as a kernel, the training can be reduced to finding a set of static, linear weights for the readout, which can be easily found by using linear regression. At its simplest, linear regression is used to predict the "dependant" variable, y, by fitting a linear equation to the provided data for the "independent" variable, ε. However, here we use multiple regression between multiple independent variables, $\boldsymbol{\varepsilon}$, and one dependant variable, y_j for the j^{th} video frame (observation), as in the following equation

$$y_j = w_0 + \sum_{i=1}^{N} w_i \varepsilon_i, \tag{3}$$

where w_0 is the intercept and w_i is the regression coefficient, or weight, for strain ε_i where i is the number of a particular readout (Fig. 2b). Estimated weights ($\hat{\boldsymbol{w}}$) are found by minimising the sum of squared residuals, SSR, calculated with the following equation

$$SSR = \sum_{j=1}^{M} (y_j - \hat{y}_j)^2, \tag{4}$$

where y_j is the target value, $\hat{y}_j$ is the predicted value and $y_j - \hat{y}_j$ is the residual of the j^{th} observation. The interpretation is that we are minimising the distance between y_j and the hyperplane $\hat{y}_j = \boldsymbol{\varepsilon}_j^T \hat{\boldsymbol{w}}$. Simply put, we try to find the optimal weights, $\hat{\boldsymbol{w}}$, that minimise the SSR.

We used Python's Scikit-Learn [12] linear model to perform multiple regression. The time series for each object position was separated into training (1400 frames) and testing data (600 frames). The first 92 frames of the training data were removed to eliminate the transients (*washout period*) caused by the initial actuation impulse [6]. The training portions were then labelled with the object position. Finally, multiple regression was used to determine a fixed set of weights ($\hat{\boldsymbol{w}}$) that will be used to predict all positions.

The predicted object position, $\hat{y}$, is then the average $\hat{y}_j$– i.e. the prediction for the j^{th} frame–over the 600 frames in the test data. The mean squared error (MSE) with respect to $\hat{y}$ is then

$$MSE = \frac{1}{M} \sum_{j=1}^{M} (y_j - \hat{y}_j)^2, \tag{5}$$

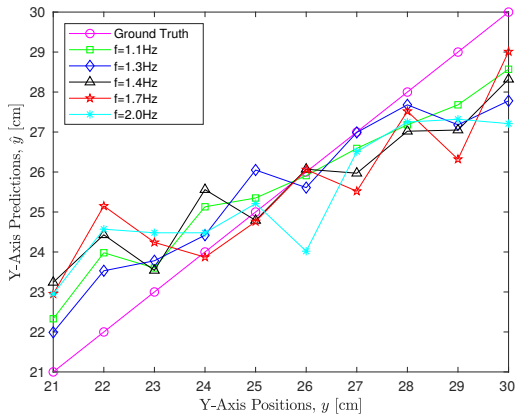
where y_j is the target value and $\hat{y}_j$ is the predicted value for the j^{th} frame and M is the number of frames in the test data.

III. RESULTS

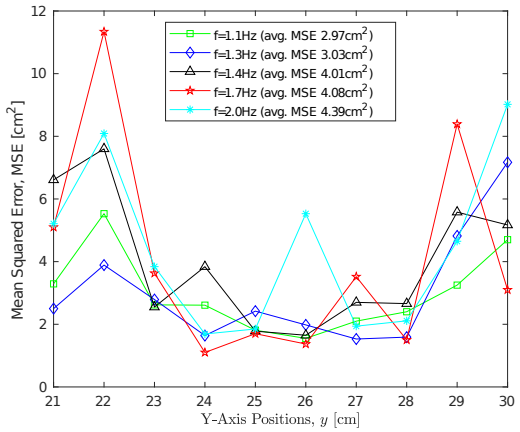
A. PREDICT OBJECT POSITION

We want to determine if the object position can be predicted using regression analysis and whether the body can be used as a computational resource for this application. A total of 10 positions were recorded on the y-axis ranging from 21cm to 30cm in steps of 1cm measured from the proximal end of the tentacle with a constant x-axis position of 20cm (Fig. 2b). The effect of the input frequency and amplitude to the driving motor of the linear slider on the accuracy of the predictions was also important to explore, but this was limited to 5 different frequencies to reduce the magnitude of the data acquisition task. These frequencies were 1.1Hz, 1.3Hz, 1.4Hz, 1.7Hz, and 2Hz at an amplitude of 4.1cm. The videos were separated into training data (1308 frames) and test data (600 frames). Therefore, the tentacle completed 19.8 (1.1Hz), 16.8 (1.3Hz), 15.6 (1.4Hz), 12.8 (1.7Hz), and 10.9 (2Hz) cycles for the training period and 9.1 (1.1Hz), 7.7 (1.3Hz), 7.1 (1.4Hz), 5.9 (1.7Hz), and 5 (2Hz) cycles for the test period.

For each of the five frequencies we learnt one set of output weights and this set of output weights was used to make predictions for all 10 object positions (Fig. 3). The predictions best fit the ground truth at the lower range of



(a) Predictions



(b) Mean Squared Error

Fig. 3. (a) Prediction performance was compared for 5 input frequencies with a constant input amplitude of 4.1cm. (b) The mean squared error was generally worse for higher input frequencies.

investigated frequencies. Additionally, all models in Fig. 3a show a characteristic overestimation of low y-axis positions and an underestimation of high y-axis positions. The highest mean squared errors (MSE) were observed for object positions at either end of the explored range on the y-axis. Further, the average MSE was 2.97cm^2, 3.03cm^2, 4.01cm^2, 4.08cm^2, and 4.39cm^2 for input frequencies 1.1Hz, 1.3Hz, 1.4Hz, 1.7Hz, and 2Hz respectively (Fig. 3b).

B. CONFUSION MATRIX

The previous results (Fig. 3) showed that the predictions, $\hat{y}$, did not monotonically increase with the ground truth positions, y. This suggests that more than one environmental state could produce the same measured behaviour. To explore this a little further, a new model was trained for each of the 10 object positions presented previously, i.e. 10 different sets of weights, one for each position, using the $\langle$4.1cm, 1.1Hz$\rangle$ pair. However, this time the model was used to classify between an object being present in the environment and no object present in the environment. Therefore, the models were trained with data from two cases, one when no object is present and one when the object is at one of the

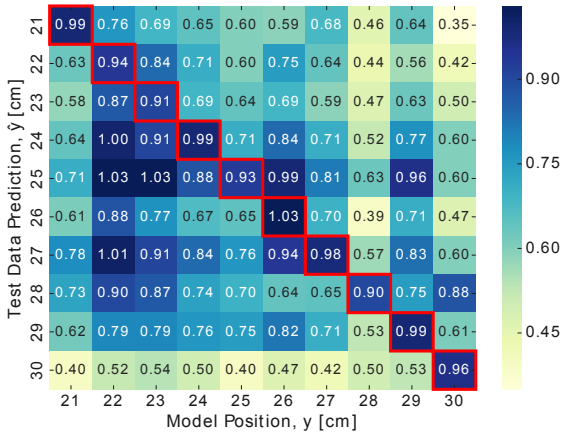


Fig. 4. Weights for 10 models (columns) were trained with data from each of the 10 object positions. Each model was then tested with data from the same object position they were trained with (diagonal). The models were also tested with data from the remaining 9 object positions (rows). Predictions were weights in the range [0, 1] where an object was classified as present if the weight was 0.5 or above. Confusion occurred most notably in models y=22cm and y=23cm where measured behaviour from some of the remaining 9 positions produced a higher weight than the measured behaviour of the object position used to train the model.

10 positions along the y-axis. The output from the models was a weight in the range [0, 1] used to classify whether the object was present or not, a weight equal or above 0.5 would be classified as present and a weight below 0.5 would be classified as not present.

A confusion matrix was made that shows the output of each model when given test data from each of the 10 object positions (Fig. 4). The weights on the diagonal (top-left to bottom-right) shows predictions when the test data was from the same object position as the model's training data. The weights on the diagonal were generally found to be the highest with some notable exceptions in models y=22cm and y=23cm where data from other object positions produced similarly high or higher weights.

C. FREQUENCY-AMPLITUDE PAIRS

We further explore the effect of the input frequency and amplitude on predictions for positions that varied in both the x and y axes. We chose 9 positions including 3 positions on the x axis (19cm, 21cm, and 23cm) and 3 positions on the y axis (10cm, 20cm and 30cm). The 25 amplitude-frequency pairs included 5 frequencies (1.1Hz, 1.3Hz, 1.4Hz, 1.7Hz, and 2Hz) and 5 amplitudes (3.56cm, 3.93cm, 4.27cm, 4.63cm, and 4.99cm). Again, predictions here are binary. An object in the environment was present if the prediction was greater than or equal to 0.5.

Models were trained using multiple linear regression. Videos for each position were separated into training (2908 frames) and testing data (600 frames). Each cell in Fig. 5 shows the average of the weights for each of the 9 positions tested. The highest and lowest average weight for all 9 positions were found for the $\langle$3.56cm, 1.3Hz$\rangle$ and $\langle$4.99cm, 2Hz$\rangle$ amplitude-frequency pairs respectively. The weights

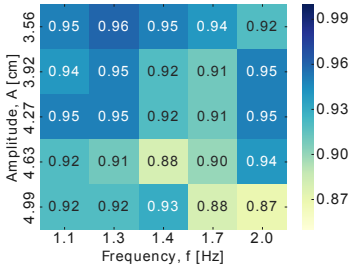


Fig. 5. Each cells shows the average weight of all 9 object positions. The ⟨3.56cm, 1.3Hz⟩ pair resulted in the highest average prediction of all 25 amplitude-frequency pairs explored while the ⟨4.99cm, 2Hz⟩ pair resulted in the lowest average prediction.

did not monotonically decrease as the object was moved away from the tentacle on the x-axis and towards the distal tip of the tentacle on the y-axis for any of the 25 pairs. We had expected to see a monotonic decrease as the object would have had a lower effect on the tentacle's behaviour as it got further away and towards the proximal end. Additionally, all predictions are correctly classified with the lowest single weight of 0.74.

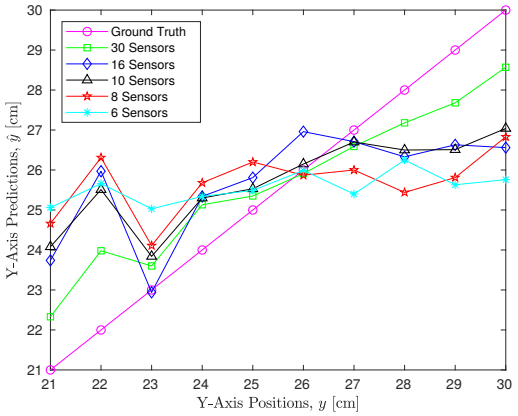
D. REDUCED SENSOR DIMENSIONALITY

In this section we explore the robustness of the proposed approach. We systematically reduced the dimensionality of the readout by removing visual strain sensors but keeping the same sensor positions. Four new models were trained using every second, third, fourth, and fifth strain sensor on either side of the tentacle and removing the rest. This corresponds to 16, 10, 8, and 6 total strain sensors respectively.

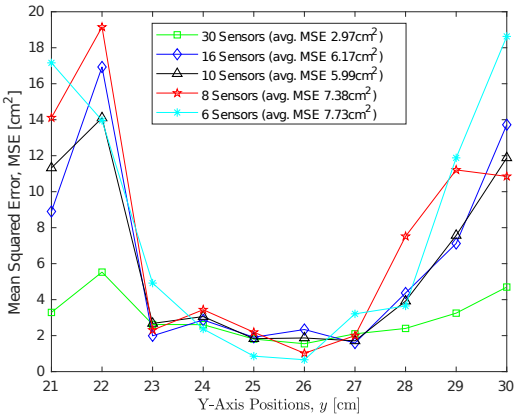
The remaining strain sensors in each of the four cases were used to make four new multiple regression models. The sensor measurements used were from the data for the ⟨4.1cm, 1.1Hz⟩ pair (Fig. 3a) as it was found to produce the best predictions. The results are summarised in Fig. 6. As expected they show that predictions worsen as the sensor dimensionality reduces. However, the performance degrades gracefully. This suggests that a reduced number of sensors, e.g. due to fail functions, can be counteracted by relearning the predictions with the reduced number of available sensors. The MSE was greatest for the highest and lowest y-axis positions (Fig. 6b). The average MSE generally increased as the number of sensors decreased. The average MSE was 2.97cm^2, 6.17cm^2, 5.99cm^2, 7.38cm^2, and 7.73cm^2 for 30, 16, 10, 8, and 6 strain sensors respectively. The average MSE was therefore more than doubled as the number of sensors was reduced from 30 to 16. However, 16 sensors still resulted in a reasonable capacity to make predictions.

IV. DISCUSSION

The experimental results in this paper support the hypothesis that the position of an object can be predicted by measuring the body dynamics of a compliant tentacle without the tentacle and object touching. Object positions along the y-axis could be predicted with a variety of motor input frequencies although low input frequencies were



(a) Predictions



(b) Mean Squared Error

Fig. 6. (a) Four new weight sets were found using every second (16 sensors), third (10 sensors), fourth (8 sensors) and fifth (6 sensors) strain sensor reading on either side of the tentacle and the ⟨4.1cm, 1.1Hz⟩ pair. (b) Reducing the number of strain sensors significantly increased the mean squared error of the models.

found to exhibit improved performance. The average MSE monotonically decreased as frequency decreased with the lowest frequency having the lowest average MSE. This suggests continuing to reduce the frequency may improve the performance further. However, we expect the existence of a lower limit for this reduction. This will be investigated in future experiments. Increasing the prediction accuracy by lowering the frequency was perhaps a result of reducing the power transmitted by the propagating wave along the tentacle as the frequency decreased. This decreased power would have had a proportionally lower effect on the overall behaviour while the vortices generated by the interaction between the tentacle and the object through the fluid would have a higher effect.

There was a persistent pattern of overestimation for low y-axis position predictions and underestimation for high y-axis position predictions for all frequencies. This was also observed in the higher MSE at both ends of the ground truth range. This suggests a polynomial, rather than a linear function, may fit the data better.

Initial results show that the body as a computational

resource can provide reasonable predictions for the position of an object in the environment. However, confusion between positions was found to occur where similar measured behaviour resulted from different object positions when using the simple binary output models. This confusion was primarily when the object was at the lower end of the explored y-axis range (y=22cm and y=23cm). An object positioned at y=30cm was close to the distal tip of the tentacle when the tentacle was at rest while y=0cm was close to the proximal end of the tentacle attached to the linear slider. Less confusion occurred as the object moved towards the distal tip of the tentacle (y=30cm). This is perhaps because the local behavioural change caused by the object affects only a relatively small area at the tip of the tentacle while the rest of the tentacle behaved as though no object was present. When the object is further towards the proximal end of the tentacle, the local behavioural change causes behavioural change all the way down the tentacle as the travelling wave is influenced. Perhaps the tentacle's larger diameter towards the proximal end causes the sensitivity to reduce. Alternatively, it may be a result of the amplitude and frequency of the wave travelling down the tentacle.

All amplitude-frequency pairs successfully predicted the presence of an object by a significant margin while further exploring how input amplitude and frequency affect classification of objects that move along both x and y axes. However, low amplitude-low frequency pairs were found to provide better predictions for all 9 positions. The $\langle$3.56cm, 1.3Hz$\rangle$ pair was found to be the best overall from the investigated parameters. Therefore, we may observe improved predictions if the frequency and amplitude are further reduced.

Interestingly, predictions did not monotonically decrease as the object was placed further away from the tentacle. This was observed for all amplitude-frequency pairs. The position furthest from the tentacle and at the distal tip (x=23cm and y=30cm) would presumably cause the least behavioural change in the tentacle but was still predicted with a comparably high weight. This suggests that a single amplitude-frequency pair is suitable for making predictions within the explored range of positions. It also shows that a subtle change in measured behaviour may be sufficient to make classifications. This approach could therefore be used to classify other changes in the environment such as the change in object size or shape. However, fewer sensors would significantly reduce the performance of these predictions. Homogeneously reducing the number of strain sensors from 30 to 16 doubled the average MSE. Fewer sensors both reduces the behavioural information and reduces the reservoir's computational performance as higher dimensional space is not as fully explored to isolate behaviour. It would be interesting to explore, in future work, how the removal of individual sensors would affect the overall performance, i.e. to see which sensors are more significant for the task. On the other hand, this also suggests that increasing the number of linear readouts from the reservoir could increase the predictive performance.

V. CONCLUSION

This paper presented a case study for how the theory of embodied intelligence can be exploited to develop a novel approach to proprioceptive sensing in soft robotics. In particular, it is shown that the position of an object can be predicted by measuring the body dynamics of a compliant robot appendage in water without direct physical contact. Furthermore, the proposed reservoir computing approach can utilize the nonlinear and complex dynamics of the compliant body to perform computations necessary for making predictions. Although we still must explore the robustness of this approach further and improve the accuracy of the predictions, the experimental results are encouraging. Future work will focus on developing novel proprioceptive sensors that can be used in high numbers but do not reduce the compliance of the tentacle. Such sensors need to be low cost to manufacture. This can be facilitated by morphological computation which does not require highly accurate sensors or the spatial relationship between sensors to be known.

VI. ACKNOWLEDGEMENTS

This work was partially supported by the EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE) at the Bristol Robotics Laboratory. Hauser was supported by Leverhulme Trust RPG-2016-345. Rossiter was supported by EPSRC grants EP/M020460/1 and EP/M026388/1 and by the Royal Academy of Engineering as a Chair in Emerging Technologies.

REFERENCES

[1] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics," *Science*, vol. 318, no. 5853, pp. 1088–1093, 2007.

[2] F. Corucci, M. Calisti, H. Hauser, and C. Laschi, "Novelty-based evolutionary design of morphing underwater robots," in *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*. ACM, 2015, pp. 145–152.

[3] F. Iida and R. Pfeifer, "Sensing through body dynamics," *Robotics and Autonomous Systems*, vol. 54, no. 8, pp. 631–640, 2006.

[4] M. Lungarella and O. Sporns, "Mapping information flow in sensorimotor networks," *PLoS Comput Biol*, vol. 2, no. 10, p. e144, 2006.

[5] R. Pfeifer and J. Bongard, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

[6] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, "Towards a theoretical foundation for morphological computation with compliant bodies," *Biological Cybernetics*, vol. 105, no. 5, pp. 355–370, 2011.

[7] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, "The role of feedback in morphological computation with compliant bodies," *Biological Cybernetics*, pp. 1–19, 2012.

[8] H. Hauser, R. M. Füchslin, and K. Nakajima, "Morphological computation - the physical body as a computational resource," in *Opinions and Outlooks on Morphological Computation*, R. M. Hauser, Helmut; Füchslin and R. Pfeifer, Eds., 2014, ch. 20, pp. 226–244.

[9] K. Nakajima, T. Li, H. Hauser, and R. Pfeifer, "Exploiting short-term memory in soft body dynamics as a computational resource," *Journal of The Royal Society Interface*, vol. 11, no. 100, p. 20140437, 2014.

[10] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Exploiting the dynamics of soft materials for machine learning," *Soft Robotics*, 2018.

[11] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.